

**UNITED STATES PATENT APPLICATION**

of

**Sharad Mathur**

and

**Gregory P. Baribault**

for

**COMPRESSING MESSAGES  
ON A PER SEMANTIC COMPONENT BASIS  
WHILE MAINTAINING A DEGREE OF HUMAN READABILITY**

## **BACKGROUND OF THE INVENTION**

### **1. The Field of the Invention**

The present invention relates to the field of compression technology. More specifically, the present invention relates to the compression of text messages on a per semantic component basis while maintaining a degree of human readability in the messages.

### **2. Background and Related Art**

Computing devices take a wide variety of forms. An early computer is often termed a "mainframe" computer system. These mainframe computers performed relatively simple processing tasks by today's standards, yet they were useful given the alternatives available at that time. One disadvantage of the mainframe computer was its size and expense. Often mainframe computers occupied entire rooms and were so expensive that users were often forced to share time on the computer with many other users. Computers were, at that time, a scarce commodity.

As semiconductor processing technology developed, the personal desk-top computer became widely available. Desk-top computers are still available as one of many alternative computing devices available to consumers. Desk-top computers are advantageous over early mainframe computers because of their reduced size and expense. In addition, despite the reduced size and expense, the processing and memory power of conventional desk-top personal computers is much greater than their larger, more expensive, mainframe ancestors.

Currently, there are a wide variety of powerful computing devices available on the market. For example, lap-top computers, cellular and digital phones, personal digital



## SUMMARY OF THE INVENTION

In accordance with the present invention, a text message is compressed on a per semantic component basis. Examples of text messages include e-mail messages, task messages, meeting request/reminder messages, and meeting summary messages. Each of these messages has logically distinct semantic components such as header fields. For example, these header fields may include the message sender, message subject, and delivery time and date, among others. In addition, the body of an e-mail message may include a current message and a number of embedded messages that document a message exchange history that led to the current message. The current message and the embedded messages may each be considered semantic components of an e-mail message.

The text message is parsed into its corresponding semantic components. Parsing of header components in the current message may be accomplished by searching for well-known characteristics of headers. On the other hand, parsing of the message body into a current message and embedded messages is more complex. First, the parser scans the text message from the bottom up and finds a common prefix of non-alphanumeric characters among contiguous lines in the message body. Three or more contiguous lines that have a common prefix are determined to be a division that is a candidate for having one or more embedded messages contained therein. If the common prefix is not consistent throughout the contiguous lines, secondary confirmation is sought that would increase the probability that the division includes one or more embedded messages.

For each division, the parser then searches from top to bottom to identify any separators, such as "----- Original Message -----" that characteristically separates embedded messages. Specifically, for each line in the division, three or more contiguous non-alphanumeric characters are sought followed by text followed by three or more

1 contiguous non-alphanumeric characteristics. If the candidate separator is a close, but  
2 inexact, match to these criteria, the separator may be classified as a weak separator in  
3 which case secondary confirmation may be sought. In addition, the parser may also look  
4 for other characteristics of a message break even if there are no separators. For example,  
5 the parser may search for two or more contiguous header lines and determine that a  
6 message break occurs immediately prior to the first header line even though there was no  
7 separator.

8       Once the message is parsed into embedded messages, all of the header lines for the  
9 embedded message are removed except for one that intuitively identifies the embedded  
10 message. For example, all headers but the From header may be removed from each of the  
11 embedded messages.

12       For at least some of the constituent semantic components, a compression method is  
13 then identified for use in compressing the semantic component. Then, the compressed  
14 component is included into the compressed text message.

15       Initially, all of the essential semantic components are compressed and truncated if  
16 they exceed a certain length. If the semantic component is a natural language component,  
17 they are compressed by natural language compression at a given level. If there is still  
18 room below the size limit. The essential semantic components are not truncated. If there  
19 is still room, optional semantic components are added. If there is still room, the level of  
20 natural language compression is reduced.

21       Reducing the level of natural language compression may be accomplished by  
22 determining the character length of the message if compressed in accordance with a first  
23 set of compression rules. If there is more room, then the character length of the message is  
24 then determined if the message is compressed using a more lenient set of rules. The

1 character length determinations are continued for increasingly more lenient compression  
2 rules until the message size finally exceeds a maximum size limit. At this stage, the  
3 method compresses the message using a set of rules that is at least as rigid as the last set of  
4 rules that resulted in the message size being within the maximum limit, but that is not as  
5 rigid as the set of rules that would result in an oversized message.

6 The principles of the present invention allow for the compression of a text message  
7 where the compression performed may be different from semantic component to semantic  
8 component. Thus, unique features, the degree of importance, the context, and language  
9 expressions of the semantic component are taken into consideration when adopting rules  
10 for compressing the semantic component. The end result is a much clearer and intuitive  
11 text compression for a given number of characters as compared to compressing the text  
12 message as a monolithic text unit.

13 Additional features and advantages of the invention will be set forth in the  
14 description which follows, and in part will be obvious from the description, or may be  
15 learned by the practice of the invention. The features and advantages of the invention may  
16 be realized and obtained by means of the instruments and combinations particularly  
17 pointed out in the appended claims. These and other features of the present invention will  
18 become more fully apparent from the following description and appended claims, or may  
19 be learned by the practice of the invention as set forth hereinafter.

## BRIEF DESCRIPTION OF THE DRAWINGS

In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 illustrates an exemplary system that provides a suitable operating environment for the present invention;

Figure 2 is a schematic diagram of an architecture that performs text compression on a per semantic component basis in accordance with the present invention;

Figure 3 is a flowchart diagram of a method of compressing text on a per semantic component basis in accordance with the present invention;

Figure 4 is a flowchart diagram of a method of identifying a compression method to be used when compressing a semantic component; and

Figure 5 is a flowchart diagram of a method of parsing a message body in accordance with the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

The present invention extends to methods, systems, and computer program products for compressing text messages on a per semantic component basis while retaining a degree of human readability. Examples of text messages that have separate semantic components include e-mail messages, task messages, meeting request/reminder messages, and meeting summary messages.

A semantic component of these messages might include any logically distinct component of the message such as, for example, a header field in the current message, or body components of the message. For example, in an e-mail message, the To, From, Date and Subject headers of the current message may be individual semantic components. The body of the message may also have distinct components. For example, a typical e-mail message may include not just the most current message, but also several embedded messages that document the history of e-mail exchanges related to the current message. The current message and each of the embedded messages may each be considered to be semantic components of the e-mail message. In one example, the headers corresponding to the embedded messages are removed from each embedded message except for one (e.g., the From header) that intuitively identifies the embedded message. The remaining portions of each embedded message are then considered a semantic component. One aspect of the present invention relates to the parsing of the message body to obtain the current message and each of the embedded messages.

In order to compress the text message, the text message is parsed into its semantic components. For at least some of the semantic components in the text message, a compression method is identified for use in compressing the semantic component. Once



1 compression is complete, the compressed semantic component is included in the  
2 compressed text message.

3 The embodiments of the present invention may comprise a special purpose or  
4 general purpose computer including various computer hardware, as discussed in greater  
5 detail below. Embodiments within the scope of the present invention also include  
6 computer-readable media for carrying or having computer-executable instructions or data  
7 structures stored thereon. Such computer-readable media can be any available media  
8 which can be accessed by a general purpose or special purpose computer.

9 By way of example, and not limitation, such computer-readable media can  
10 comprise physical computer-readable media such as RAM, ROM, EEPROM, CD-ROM or  
11 other optical disk storage, magnetic disk storage or other magnetic storage devices, or any  
12 other medium which can be used to carry or store desired program code means in the form  
13 of computer-executable instructions or data structures and which can be accessed by a  
14 general purpose or special purpose computer.

15 When information is transferred or provided over a network or another  
16 communications connection (either hardwired, wireless, or a combination of hardwired or  
17 wireless) to a computer, the computer properly views the connection as a computer-  
18 readable medium. Thus, any such connection is properly termed a computer-readable  
19 medium. Combinations of the above should also be included within the scope of  
20 computer-readable media. Computer-executable instructions comprise, for example,  
21 instructions and data which cause a general purpose computer, special purpose computer,  
22 or special purpose processing device to perform a certain function or group of functions.

23 Figure 1 and the following discussion are intended to provide a brief, general  
24 description of a suitable computing environment in which the invention may be

1 implemented. Although not required, the invention will be described in the general context  
2 of computer-executable instructions, such as program modules, being executed by  
3 computers in network environments. Generally, program modules include routines,  
4 programs, objects, components, data structures, etc. that perform particular tasks or  
5 implement particular abstract data types. Computer-executable instructions, associated  
6 data structures, and program modules represent examples of the program code means for  
7 executing steps of the methods disclosed herein. The particular sequence of such  
8 executable instructions or associated data structures represents examples of corresponding  
9 acts for implementing the functions described in such steps.

10 Those skilled in the art will appreciate that the invention may be practiced in  
11 network computing environments with many types of computer system configurations,  
12 including personal computers, hand-held devices, multi-processor systems,  
13 microprocessor-based or programmable consumer electronics, network PCs,  
14 minicomputers, mainframe computers, and the like. The invention may also be practiced  
15 in distributed computing environments where tasks are performed by local and remote  
16 processing devices that are linked (either by hardwired links, wireless links, or by a  
17 combination of hardwired or wireless links) through a communications network. In a  
18 distributed computing environment, program modules may be located in both local and  
19 remote memory storage devices.

20 With reference to Figure 1, an exemplary system for implementing the invention  
21 includes a general purpose computing device in the form of a conventional computer 120,  
22 including a processing unit 121, a system memory 122, and a system bus 123 that couples  
23 various system components including the system memory 122 to the processing unit 121.  
24 The system bus 123 may be any of several types of bus structures including a memory bus

1 or memory controller, a peripheral bus, and a local bus using any of a variety of bus  
2 architectures. The system memory includes read only memory (ROM) 124 and random  
3 access memory (RAM) 125. A basic input/output system (BIOS) 126, containing the basic  
4 routines that help transfer information between elements within the computer 120, such as  
5 during start-up, may be stored in ROM 124.

6 The computer 120 may also include a magnetic hard disk drive 127 for reading  
7 from and writing to a magnetic hard disk 139, a magnetic disk drive 128 for reading from  
8 or writing to a removable magnetic disk 129, and an optical disk drive 130 for reading  
9 from or writing to removable optical disk 131 such as a CD-ROM or other optical media.  
10 The magnetic hard disk drive 127, magnetic disk drive 128, and optical disk drive 130 are  
11 connected to the system bus 123 by a hard disk drive interface 132, a magnetic disk drive-  
12 interface 133, and an optical drive interface 134, respectively. The drives and their  
13 associated computer-readable media provide nonvolatile storage of computer-executable  
14 instructions, data structures, program modules and other data for the computer 120.  
15 Although the exemplary environment described herein employs a magnetic hard disk 139,  
16 a removable magnetic disk 129 and a removable optical disk 131, other types of computer  
17 readable media for storing data can be used, including magnetic cassettes, flash memory  
18 cards, digital versatile disks, Bernoulli cartridges, RAMs, ROMs, and the like.

19 Program code means comprising one or more program modules may be stored on  
20 the hard disk 139, magnetic disk 129, optical disk 131, ROM 124 or RAM 125, including  
21 an operating system 135, one or more application programs 136, other program modules  
22 137, and program data 138. A user may enter commands and information into the  
23 computer 120 through keyboard 140, pointing device 142, or other input devices (not  
24 shown), such as a microphone, joy stick, game pad, satellite dish, scanner, or the like.

1 These and other input devices are often connected to the processing unit 121 through a  
2 serial port interface 146 coupled to system bus 123. Alternatively, the input devices may  
3 be connected by other interfaces, such as a parallel port, a game port or a universal serial  
4 bus (USB). A monitor 147 or another display device is also connected to system bus 123  
5 via an interface, such as video adapter 148. In addition to the monitor, personal computers  
6 typically include other peripheral output devices (not shown), such as speakers and  
7 printers.

8 The computer 120 may operate in a networked environment using logical  
9 connections to one or more remote computers, such as remote computers 149a and 149b.  
10 Remote computers 149a and 149b may each be another personal computer, a server, a  
11 router, a network PC, a peer device or other common network node, and typically include  
12 many or all of the elements described above relative to the computer 120, although only  
13 memory storage devices 150a and 150b and their associated application programs 136a and  
14 136b have been illustrated in Figure 1. The logical connections depicted in Figure 1  
15 include a local area network (LAN) 151 and a wide area network (WAN) 152 that are  
16 presented here by way of example and not limitation. Such networking environments are  
17 commonplace in office-wide or enterprise-wide computer networks, intranets and the  
18 Internet.

19 When used in a LAN networking environment, the computer 120 is connected to  
20 the local network 151 through a network interface or adapter 153. When used in a WAN  
21 networking environment, the computer 120 may include a modem 154, a wireless link, or  
22 other means for establishing communications over the wide area network 152, such as the  
23 Internet. The modem 154, which may be internal or external, is connected to the system  
24 bus 123 via the serial port interface 146. In a networked environment, program modules

1 depicted relative to the computer 120, or portions thereof, may be stored in the remote  
2 memory storage device. It will be appreciated that the network connections shown are  
3 exemplary and other means of establishing communications over wide area network 152  
4 may be used.

5 Figure 2 schematically illustrates an architecture 200 in accordance with the present  
6 invention that performs text compression on a per component basis. Among other things,  
7 the architecture includes a message compressor 201 which receives a text message 202  
8 having several semantic components. The message compressor 201 optionally uses a  
9 natural language component compressor 206 to form the compressed text message 211.  
10 The message compressor 201, the text message 202, the natural language component  
11 compressor 206 and the compressed text message 211 may comprise, for example, part of  
12 the program code means described above with reference to Figure 1.

13 Figure 3 illustrates a flowchart of a method 300 for compressing the text message  
14 on a per semantic component basis in accordance with the present invention. The method  
15 300 of Figure 3 may be performed by the architecture 200 of Figure 2. Accordingly, the  
16 description of the architecture 200 and of the method 300 will be interweaved with  
17 frequent reference to both Figure 2 and Figure 3.

18 In method 300, the text message is accessed (act 301). This is illustrated in Figure  
19 2 by message compressor 201 receiving the text message 202. The method compresses  
20 text messages on a per semantic component basis when a text message includes more than  
21 one semantic component. For example, the text message 202 includes three semantic  
22 components including component A (element 203), component B (element 204), and  
23 component C (element 205). Although these semantic components are represented  
24

1 generically in the text message 202 as elements 203-205, various specific examples of text  
2 messages and associated semantic components will be described herein.

3 As an initial example used to illustrate the principles of the present invention,  
4 suppose that the text message 202 is an e-mail message. Typical semantic components of  
5 an e-mail message would include a To header identifying the recipient of the message, a  
6 From header identifying the sender of the message, a Date header identifying the date and  
7 possibly the time of day that the message was sent, and a Subject header summarizing the  
8 content of the message. In addition, the current message in the body as well as any  
9 embedded messages may each be a semantic component of the text message.

10 After accessing the text message (act 301), the message compressor 201 parses the  
11 text message into its various semantic components (act 302). For example, a parsing  
12 component such as parser 215 may be used by the message compressor 201 to parse the  
13 various semantic components of the text message 202 including components A, B and C.  
14 Parsing of header fields is handled by searching for well-known characteristics of header  
15 fields. However, parsing of a body of a text message is more complex especially when a  
16 thread of e-mail messages is embedded in the body of an e-mail message. The parsing of  
17 the body of an e-mail message will be described in further detailed below.

18 After parsing the text message into semantic components (act 302), the method  
19 includes a step for optimizing the text compression on a per semantic component basis so  
20 that the more important information is included in the compressed text message (step 303).

21 Step 303 includes various corresponding acts 304 through 305, which will now be  
22 described. First, a compression method is identified for use when compressing the  
23 semantic component for inclusion in the compressed message (act 304). For example, the  
24 identified compression may include a specific level of compression performed on natural

language semantic components, a custom compression technique, or a combination of both. It may even be that no compression at all needs to be performed on the semantic component. The compressed semantic component (or perhaps the entire original semantic component if it is not to be compressed) is then included in the compressed text message (act 305). For example, compressed text message 211 generated by the message compressor 201 includes compressed components 212, 213 and 214 corresponding to components 203, 204 and 205 in the original text message 202.

Now described is the method for determining the level of compression to be performed on the semantic component. As mentioned above, examples of text message types that include semantic components include e-mail messages, task messages, meeting request/reminder messages, and meeting summary messages. Each of these text message types is listed in the following Table 1 along with corresponding semantic components.

Text Message Type	Semantic Component	Essential?
E-mail Message	Subject Header	Essential
	From Header	Essential
	Date Header	Optional
	Body (Current Message)	Essential
	Body (Last Embedded Message)	Essential
	Body (Other Embedded Messages)	Optional
Task Message	Subject Header	Essential
	From Header	Essential
	Start Date Header	Essential
	Due Date Header	Essential
	Percent Complete Header	Essential
	Body (Current Message)	Essential
	Body (Last Embedded Message)	Essential
	Body (Other Embedded Messages)	Optional
Meeting Request/Reminder Message	Subject Header	Essential
	Date Header	Essential
	Location Header	Essential
	From Header	Optional
	Body (Current Message)	Essential
	Body (Last Embedded Message)	Essential
	Body (Other Embedded Messages)	Optional
Meeting Summary Message	Subject Header	Essential
	Time Header	Essential
	Location Header	Essential
	Tasks	Essential
	Subject	Essential

Table 1



1 Referring to Table 1, the semantic components for each message type are identified  
2 as being “essential” or “optional”. The essential semantic components correspond to those  
3 semantic components that are most important to be included in the compressed text  
4 message. The optional semantic components correspond to those semantic components  
5 that are less important, but are desirable to include if there is enough room.

6 For example, for an e-mail message, it is essential that the Subject header be  
7 included since it is helpful in the essential function of summarizing the content of the e-  
8 mail message. The From header is essential since it identifies the sender, which is critical  
9 for a reader in evaluating the context and importance of an e-mail message. The current  
10 message in the body is essential because it is the primary content that is intended to be  
11 conveyed by an e-mail message. Also, the last embedded message in the body is essential  
12 since it provides critical context information in cases where the current message is a reply  
13 to another message. However, any earlier embedded messages are only optional since it is  
14 thought that the current message and the last embedded message should be sufficient to  
15 derive most of the context of the current message. The Date header is also considered  
16 optional since it is not critical to know to understand the e-mail message. Any semantic  
17 components that are not considered optional or essential, such as the To header in an e-  
18 mail message, are not illustrated in Table 1 since they are not to be included in the  
19 compressed text message.

20 Some semantic components include natural language text that includes language as  
21 a person would naturally speak it. For example, the body, subject, and task descriptions of  
22 a message typically include natural language expressions. On the other hand, headers  
23 other than the subject typically do not include natural language expressions. The natural  
24 language components of the text message may be compressed by a natural language text

1 component compressor 206 (see Figure 2). An example of a natural language component  
2 compressor is described in co-pending and commonly-owned United States patent  
3 application serial number [not yet assigned] filed 26 January 2001 and entitled  
4 "Linguistically Intelligent Text Compression", which is incorporated herein by reference in  
5 its entirety.

6 When provided to natural language component compressor 206, the natural  
7 language component compressor 206 returns the natural language component in four  
8 different versions 207 through 210, each version representing a different level of  
9 compression which will be called herein level 1 compression, level 2 compression, level 3  
10 compression, and level 4 compression, respectively, in order of increasing compression.  
11 Level 0 compression is used to refer to no natural language compression at all. One of  
12 these versions of the natural language component may then be selected for inclusion in the  
13 compressed text message.

14 The type of compression that is to be performed on the semantic component  
15 depends on the type of the semantic component. The compression that may be performed  
16 will now be described for each semantic component listed in Table 1.

17 The Subject header is common throughout all the text message types and is  
18 considered to be a natural language component. Accordingly, the Subject header  
19 component of any one of the text message types is passed to the natural language  
20 compressor as represented in Figure 2 by the message compressor 201 passing component  
21 C (element 205) to the natural language component compressor 206. The natural language  
22 component compressor 206 then returns the four compressed versions (i.e., versions 207,  
23 208, 209 and 210) back to the message compressor 201. The method for selecting one of  
24 these versions is described in further detail below with respect to Figure 4.

1 The From header is common to the E-mail Message and Task Message text  
2 message types and is not considered to be a natural language semantic component. After  
3 all, it just lists the name of the sender. A customized compression is thus performed on the  
4 From header. For example, the user may elect to have the full last name and the first  
5 initial, the full first name and then the last initial, and so forth. The user may also elect to  
6 have the device search for a phone number for that name in a database and include that  
7 phone number in the From field.

8 Time-based semantic components such as the Date header, the Start Date header,  
9 the Due Date header, and the Time header are also not considered natural language  
10 components, and are thus not passed to the natural language component compressor 206.  
11 If the time-based component includes a start time, a start date, an end time, and an end  
12 date, and if the start and end dates are the same, then the dates are left out. If the dates are  
13 different, but the year is the same, the year is left out. If the date is within the next six  
14 months, the year is left out. In short, any implicit information is stripped out of time-based  
15 semantic components.

16 As for the Location header, any Uniform Resource Locators (URLs) are stripped  
17 out if the device that is to read the text message is a device that is not capable of navigating  
18 to a URL. For example, the device that reads the text message may be a digital telephone  
19 that lacks Web navigation functionality. The rest of the Location Header may be passed to  
20 the natural language component compressor 206.

21 The Task header may be considered a natural language component and thus may be  
22 passed to the natural language component compressor 206.

1 The Percent Complete header may be compressed by changing spelled out numbers  
2 to digits representing the number. Also, the text "percent" may be changed to the symbol  
3 "%".

4 The body is parsed into one or more logical sections. For example, in an e-mail  
5 message that includes a thread documenting the history of e-mail messages, the body may  
6 be parsed into separate messages within the thread. The body is considered a natural  
7 language element and thus the parsed sections from the body are passed to the natural  
8 language component compressor 206. The mechanism for parsing the body of a message  
9 is described in further detail below.

11 Each semantic component has a specified initial maximum size. For example, the  
12 Subject header has an initial maximum size that is proportional to a parameter called herein  
13 "MAX\_SUBJECT". Each of the time-based semantic components has an initial maximum  
14 size that is proportional to a parameter called herein "MAX\_TIMERANGESIZE". The  
15 From header has an initial maximum size that is proportional to a parameter called herein  
16 "MAX\_FROMSIZE". The Location header has an initial maximum size that is  
17 proportional to a parameter called herein "MAX\_LOCATIONSIZE". The body has an  
18 initial maximum size that is proportion to a parameter called herein "MAX\_BODYSIZE".

19 If the semantic component is not compressed using the natural language component  
20 compressor 206, then the initial maximum size of the compressed component is equal to  
21 the corresponding parameter. If the semantic component is compressed using the natural  
22 language component compressor 206, then the initial maximum size of the compressed  
23 component is equal to the corresponding parameter if the natural language compression  
24 level is 0, 1 or 2. However, the initial maximum size is 80% of the corresponding

parameter if level 3 compression is employed, and 60% of the corresponding parameter if level 4 compression is employed. Any compressed text beyond the initial maximum size is initially truncated. The following may be the values of the parameters:

MAX\_FROMSIZE = 32

MAX\_LOCATIONSIZE = 40

MAX\_TIMERANGESIZE = 24

MAX\_SUBJECT = 64

MAX\_BODYSIZE = 2048

If natural language component compression is not to be performed on a semantic component, the custom compression techniques described above are employed on that semantic component. However, if natural language component compression is employed, the level of natural language compression that is performed on the natural language component is determined by constructing one or more test compressed messages.

Figure 4 illustrates a method 400 for identifying a compression method to be used for each natural language component. The method will be described with respect to an example in which the user has indicated a maximum message fragment size of 160 characters with a maximum four fragments per message. Each text message may then include 640 characters. Also assume that the user has specified a maximum natural language compression level of 3. Perhaps the user may just have difficulty understanding semantic components compressed at level 4 natural language compression. Also, assume that the text message to be compressed is an e-mail message.

1 First, the initial size of the message is determined (act 401) where the message  
2 includes only those semantic components identified as essential, and where the essential  
3 components are truncated if they exceed the initial maximum size, and where natural  
4 language compression is performed at the maximum natural language compression level.  
5 The initial size of the message may be determined by actually constructing the message  
6 and then counting the number of characters. However, a lower processing method  
7 involves determining the number of characters in each semantic component and then  
8 adding these numbers for the total size. Each semantic component listed above for an e-  
9 mail message will now be evaluated to determine how a bare minimum test message would  
10 be created in the example.

11 The Subject header is considered essential and thus at least a compressed form of  
12 the Subject header is included in the compressed e-mail message. Since natural language  
13 component compression is employed at level 3 on the Subject header, the initial maximum  
14 size of the Subject header is  $0.8 * \text{MAX\_SUBJECT}$  which is equal to  $0.8 * 64 \cong 52$   
15 characters. If the non-compressed version of the Subject header were 100 characters, the  
16 level 3 compression version of the Subject header returned by the natural language  
17 component compressor 206 may be, for example, 67 characters. The last 15 characters of  
18 the compressed Subject header would thus be truncated to create a Subject header that is  
19 just 52 characters long.

20 The From header is also considered essential and thus compression is performed as  
21 described above for the From header. Natural language component compression is not  
22 employed on the From header and thus the initial maximum size of the compressed From  
23 header is equal to  $\text{MAX\_FROMSIZE}$  or 32 characters. Suppose that the compressed From  
24

1 header size is only 14 characters long. All of the 14 characters are included in the  
2 compressed From header.

3 The Date header is only optional. Therefore, the date header does not contribute  
4 any characters to the initial size.

5 The current message and the temporally last embedded message in the body of the  
6 e-mail message are considered essential. These body components are compressed by  
7 natural language component compression. Since level 3 natural language compression is  
8 employed, the actual initial maximum size of the body is  $0.8 * \text{MAX\_BODYSIZE}$  or  
9  $0.8 * 2048 \cong 1638$  characters. Suppose the level 3 compressed form of the current message  
10 is 150 characters, and the level 3 compressed form of the last embedded message is 155  
11 characters. The essential components of the body would thus be only 305 characters long.  
12 There would be no truncation of the compressed body.

13 In this example, the initial size of the message would be 52 characters for the  
14 Subject header plus 14 characters for the From header plus 305 characters for the essential  
15 portions of the body. The total size for the initial test message would thus be 371  
16 characters. This is well within the 640 character limit set by the user. Thus, more of the  
17 text message may be sent (YES in decision block 402).

18 In order to fill this extra space, the size of the message is determined if truncation  
19 were eliminated from the message In this example, the untruncated Subject header was 67  
20 characters. On the other hand, the From header and the essential body components were  
21 within the initial maximum lengths and thus were not truncated. The size of the message  
22 would thus be increased by 15 characters to 386 characters in act 403 due to the adding  
23 back of the 15 truncated characters from the Subject header.

1 Since there is still room within the user specified maximum message size to  
2 increase the size of the test message (YES in decision block 404), the size of the message  
3 is determined if the message included all of the optional fields compressed at the  
4 maximum natural language compression level (act 405). Since this is an e-mail message,  
5 the only optional field is the Date field and any embedded messages temporally earlier than  
6 the last embedded message. Assume that there are two embedded messages in the  
7 body and that the optional embedded message is 180 characters long when compressed at  
8 level 3 compression. The date header is compressed as described above. Suppose that the  
9 compressed date is 15 characters. The message size would thus increase another 180  
10 characters for the additional embedded message in the body and increase another 15  
11 characters for the date header. The total increase by act 405 would thus be 195 characters.  
12 Thus, the size estimate would increase from 386 characters to 581 characters, still within  
13 the user specific maximum size (YES in decision block 406).

14 At this stage, the size of the message has been determined with all of the essential  
15 and optional semantic components included in the message in untruncated form and  
16 compressed at the maximum natural language compression level. Next, the compression  
17 level of each natural language oriented semantic component is reduced one at a time until  
18 either there is no longer enough room (NO in decision block 406), or the compression level  
19 has been reduced for all of the natural language oriented semantic components in the  
20 message. In the latter case, if there is still room, the compression level is further reduced  
21 one semantic component at a time until there is no longer any room available.

22 For example, there is more room available (YES in decision block 406) since the  
23 size of the message would only be 581 characters if the message was at the maximum  
24



1 natural language compression of level 3, with all semantic components (essential and  
2 optional) included and being untruncated.

3         Since there is still more room available, the compression level of one of the natural  
4 language semantic components is reduced to level 2. Suppose the Subject header is  
5 reduced to level 2 compression and that this would increase the Subject header from 67  
6 characters to 79 characters. The total size of the message would thus increase by 12 to 593  
7 characters.

8         Since there is still room, the next natural language component size is reduce to  
9 level 2. The next natural language component is the current message. Suppose reducing  
10 the compression level increases the current message from 150 characters to 185 characters.  
11 The total size of the message would thus increase by 35 to 628 characters. There is still  
12 room, albeit only for 12 more characters.

13         Suppose that reducing the compression level for the next natural language  
14 component (i.e., the last embedded message) increases the last embedded message size  
15 from 155 characters to 190 characters. The total size of the message would increase by 35  
16 to 663 characters, 23 characters more than the maximum (NO in decision block 406).

17         At this stage, it is determined what may be done to the message to reduce the size  
18 to at or below the maximum size of 640 characters (act 408). In this example, that may  
19 involve only reducing the compression level for some of the sentences in the last  
20 embedded message.

21         By gradually reducing the strictness of compression in this manner, the size of the  
22 test message is increased until the appropriate level of natural language compression is  
23 identified (act 304 of Figure 3).

1 A method of compressing a text message on a per semantic component basis has  
2 been described. A method of parsing of the body of a message to extract messages that  
3 were embedded within the message body is now described. An e-mail message may  
4 contain a thread that includes the current message and one or more embedded messages  
5 that are contain within a message body. The following characters illustrate a message  
6 body that contains two embedded messages that document a history of e-mail messages.

7  
8 25. Hello again Jack-

9 24. I'll bring the pail. But don't forget you ultra-grip running shoes. You know  
10 23. how slippery the hill can get this time of year.

11 22. Jill

12 21. <cr>

13 20. #####-----Original Message-----

14 19. #####From: Jack

15 18. #####Sent: Saturday, December 23, 2000 5:24 PM

16 17. #####To: Jill

17 16. #####Subject: RE: Hill Run

18 15. #####<cr>

19 14. #####Hello Jill-

20 13. #####Sure. Be sure to bring your pail.

21 12. #####Jack

22 11. #####<cr>

23 10. #####-----Original Message-----

24 9. #####From: Jill

```
8. #####Sent:    Saturday,December23,20005:23 PM
7. #####To:      Jack
6. #####Subject: Hill Run
5. #####<cr>
4. #####Hello Jack-
3. #####Are you still planning on taking a run up the hill? I'm
2. #####running short on water.
1. #####Jill
```

This text appears as it would in the body of an e-mail except that, for clarity, line numbers are added, space characters that introduce a line are replaced with “#” characters, and a carriage return is marked with the characters “<cr>”. In this example, the bottom-most embedded message is from Jill to Jack and is the temporally earliest embedded message in the thread. This will also be called the original embedded message and corresponds to lines 1 through 10. The embedded message above the original embedded message is from Jack to Jill and is a message that responds to the original embedded message. This will also be called the response embedded message and corresponds to lines 11 through 20. The upper-most message is the current message that responds to the response embedded message. This will be called the current message and corresponds to lines 21 through 25.

The current message and the temporally last embedded message are considered to be the essential messages in the body. Any embedded message other than the most recent embedded message is considered optional. Thus, the current message and the response

1 embedded message are considered essential, while the original embedded message is  
2 considered optional.

3 Figure 5 illustrates a method 500 of parsing the body into its component messages.  
4 First, the parser 215 locates message breaks in the body that are characteristic of  
5 separations between messages in the body (act 501). Then, the material from each  
6 message break to the next message break, if any, is identified as corresponding to an  
7 embedded message (act 502).

8 In order to identify the separators, the parser 215 first evaluates the message body  
9 in reverse, line-by-line to divide the body into divisions that are characteristic of including  
10 one or more messages. For each such division, the parser 215 then evaluates the division  
11 line-by-line from top to bottom to identify any messages within the division.

12 First, the parser 215 moves in reverse from bottom to top through the message. At  
13 the last line, the algorithm determines if there are any non-alphanumeric start characters in  
14 the line before the first alphanumeric character. The parser then moves up to the next line  
15 and determines the non-alphanumeric start characters for that line, and so on up preceeding  
16 lines. If, starting at the bottom, more than three contiguous lines are found with the same  
17 non-alphanumeric start characters (e.g., ">>"), then the parser assumes that the start  
18 characters are common line prefixes that are used to indent an embedded message. The  
19 body parser continues to preceed up the lines until a line is found that does not have the  
20 common prefix. Everything below that line is determined to be a division of the body.

21 In the above example thread, the non-alphanumeric start characters are ten space  
22 characters (each shown as "#") from line 1 until line 10. If the common prefix in the  
23 division extends down through the entire division as it does in the original embedded  
24 message of the example thread shown above, then the prefix is defined as a "strong" prefix

1 meaning that the common prefix is a strong indicator that the division includes a message.  
2 If the common prefix in the division does not extend down through the entire division, then  
3 the prefix is determined to be a "weak " prefix meaning that the common prefix is a  
4 weaker indicator that the division does indeed include one or more messages.

5 Thus, if the division has only a weak prefix, the body parser searches for a  
6 secondary confirmation that the division includes a message. Such secondary confirmation  
7 may include looking at the first line in a division to determine whether or not the line is  
8 characteristic of a mail header. For example, a mail header may be characterized by text  
9 followed by a colon followed by more text. Secondary confirmation may also include  
10 looking at the first line in a division to determine whether or not the line is characteristic of  
11 a mail separator. For example, a mail separator may be characterized by a series of the  
12 three or more of the same non-alphanumeric characters followed by text followed by a  
13 second series of three or more of the same non-alphanumeric characters. If a division has a  
14 strong prefix or a weak prefix and a secondary confirmation, then the division is  
15 determined to be characteristic of including one or more messages.

16 In the above example thread, this process would determine that the ten space  
17 characters leading off lines 1 through 10 is a strong prefix and thus the method would  
18 identify lines 1 through 10 as corresponding to a division. The method would then proceed  
19 from lines 11 through 20 to identify the five space characters leading off each lines 11  
20 through 20 is indeed a strong prefix and thus the method would identify lines 11 through  
21 20 as corresponding to a division.

22 For each division that is characteristic of including one or more messages, the body  
23 parser then removes any prefixes and evaluates that division, line-by-line, starting at the  
24 top of the division. The bottom division would then appear as follows:

10. -----Original Message-----  
9. From: Jill  
8. Sent: Saturday, #December#23, #2000#5:23 PM  
7. To: Jack  
6. Subject: Hill Run  
5. <cr>  
4. Hello Jack-  
3. Are you still planning on taking a run up the hill? I'm  
2. running short on water.  
1. Jill

The top division would appear as follows:

20. -----Original Message-----  
19. From: Jack  
18. Sent: Saturday, December 23, 2000 5:24 PM  
17. To: Jill  
16. Subject: RE: Hill Run  
15. <cr>  
14. Hello Jill-  
13. Sure. Be sure to bring your pail.  
12. Jack  
11. <cr>

1  
2 The body parser looks for a message separator or a message header which would  
3 indicate the beginning of a message. For each division, if the body parser finds a line that  
4 begins with a series of three or more of the same non-alphanumeric characters followed by  
5 text followed by a series of three or more of the same non-alphanumeric characters, then  
6 that line is determine to be a "strong" separator. However, if the line only begins with a  
7 series of three or more of the same non-alphanumeric text, then that line is determined to  
8 be a "weak" separator, and secondary confirmation is sought. Such secondary  
9 confirmation would include looking at the next line to determine if the line looks like a  
10 header line as when there is some text followed by a colon followed by more text. If a  
11 strong separator is found or if a weak separator is found with secondary confirmation, then  
12 that separator line is determined to be the beginning of a message.

13 Even if there is no separator found in the division, the body parser looks for other  
14 indications that there is a message break. For example, the body parser searches for  
15 contiguous lines that are characteristic of being a message header. A message header line  
16 may be identified by, for example, searching for a line in which there is some text followed  
17 by a colon followed by some text. If more than one contiguous line of message headers is  
18 found, then there is determined to be a message break before the first contiguous line of  
19 message headers.

20 In the above example thread, both divisions begin with a line of characters "-----  
21 Original Message-----". This is indeed a line that begins with a series of three or more of  
22 the same non-alphanumeric characters (e.g., "-") followed by text (e.g., Original Message)  
23 followed by a series of three or more of the same non-alphanumeric characters (e.g., "-").  
24 Thus, lines 10 and 20 are determined to be strong separators. There are no weak separator

1 in either division. Also, it was not necessary to search for a message break using message  
2 header lines since separators existed for each message. Thus, each division as a whole is  
3 determined to be an embedded message. The current message and each of the stripped  
4 embedded messages are then considered to be a semantic component.

5 In this manner, the body is parsed into separate message components. If the  
6 message component had a header, then the parser looks for a header that identifies the  
7 message intuitively. For example, in the English language, the parser might look for the  
8 From header. The parser would then include that From header as part of the message  
9 component and remove all other headers.

10 The present invention may be embodied in other specific forms without departing  
11 from its spirit or essential characteristics. The described embodiments are to be considered  
12 in all respects only as illustrative and not restrictive. The scope of the invention is,  
13 therefore, indicated by the appended claims rather than by the foregoing description. All  
14 changes which come within the meaning and range of equivalency of the claims are to be  
15 embraced within their scope.

16 What is claimed and desired to be secured by United States Letters Patent is:  
17  
18  
19  
20  
21  
22  
23  
24